UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/772,626	02/05/2004	Michael Kovacs	BEAS-01302US1	5661

23910 7590 08/29/2007
FLIESLER MEYER LLP
650 CALIFORNIA STREET
14TH FLOOR
SAN FRANCISCO, CA 94108

EXAMINER	
VU, TUAN A	
ART UNIT	PAPER NUMBER
2193	
MAIL DATE	DELIVERY MODE
08/29/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary

Application No. 10/772,626	Applicant(s) KOVACS ET AL.	
Examiner Tuan A. Vu	Art Unit 2193	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on 2/5/04.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) 1-28 is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) 1-28 is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on 2/5/04 is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☒ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date 8/16/07.

4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application

6) ☐ Other: _____.

DETAILED ACTION

1. This action is responsive to the application filed 2/5/2004.

Claims 1-28 have been submitted for examination.

Claim Objections

2. Claims 1, 8 are objected to because of the following informalities: the 'one deployment descriptors' appears to have a typo error in the plural 'descriptors'. Appropriate correction is required.

Double Patenting

3. The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the "right to exclude" granted by a patent and to prevent possible harassment by multiple assignees. A nonstatutory obviousness-type double patenting rejection is appropriate where the conflicting claims are not identical, but at least one examined application claim is not patentably distinct from the reference claim(s) because the examined application claim is either anticipated by, or would have been obvious over, the reference claim(s). See, e.g., *In re Berg*, 140 F.3d 1428, 46 USPQ2d 1226 (Fed. Cir. 1998); *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970); and *In re Thorington*, 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) or 1.321(d) may be used to overcome an actual or provisional rejection based on a nonstatutory double patenting ground provided the conflicting application or patent either is shown to be commonly owned with this application, or claims an invention made as a result of activities undertaken within the scope of a joint research agreement.

Effective January 1, 1994, a registered attorney or agent of record may sign a terminal disclaimer. A terminal disclaimer signed by the assignee must fully comply with 37 CFR 3.73(b).

4. Claims 5, 8, 15, 22 are provisionally rejected on the ground of nonstatutory obviousness-type double patenting as being unpatentable over claims 6, 12 of copending Application No. 11,088,173 (hereinafter '173) in view of WebLogic Server 6.1, 'Developing Weblogic Server

J2EE Applications' (hereinafter WLS_6.1) . Although the conflicting claims are not identical, they are not patentably distinct from each other because of the following observations.

Following are but a few examples as to how the certain claims from the instant invention and from the above copending application are conflicting.

As for instant claim 5, '173 claim 6 recites a descriptor representing modifiable configuration data with annotation for a bean descriptor; a descriptor class management for generating and updating one or more descriptor from the descriptor classes, a validator for validating properties of a descriptor bean at parse time; hence has taught the language conveyed by claim 5 such as 'generating a representation of a descriptor, creating a descriptor, validating one descriptor, automatically updating descriptor to reflect changes in at least one source files (i.e. *properties from a bean or class* – from '173 claim 6). But '173 claim 6 does not teach *deployment descriptor* nor does it recite 'automatically repairing a first descriptor if the first descriptor is defective' and 'deploying an application with one deployment descriptor'; however, '173 claim 6 mentions about bean class properties being validated for a implementation customizing the use of the bean in a Java implementation, and updating of the descriptor properties whenever bean descriptor is modified. It would have been obvious to provide a repairing capability to the parser and runtime implementation of the above Java bean so that the customizer of '173 can update said descriptor bean defective properties in order to implement such descriptor in a deployment environment as taught by WLS_6.1(see *Editing Application Deployment Descriptors*: pg. 16-20), thereby rendering *repairing* capability of such validator and *deployment descriptor* both obvious in view of the customizing and automatic updating in '173 claiming of a implementation runtime.

As per instant claims 15 and 22, '173 claim 12 also recites comparing first and second representation of descriptor (*versions of descriptor*) and based on some acceptance proposal, updating the descriptor with accepting such proposal, based on difference between the versions. Although the language of `173 claim 6 is not same as that of claims 15, 22, the concept for updating some descriptor being out-of-date against another version is indicated; however, `173 claim 12 does not recite deployment descriptor. However, the *deployment descriptor* limitation in view of WLS_6.1, have been rendered obvious in view of the update environment to synchronize needed Descriptor properties using`173 Descriptor class management module, as set forth above.

As per instant claim 8, '173 claim 12 also teaches 'first and second representation of one descriptor', comparing said first and second representations, automatically updating when one of such is out-of-date, according to the rationale as set forth above. Claim 12 does not recite 'deployment descriptor' nor 'automatically repairing if first deployment descriptor is defective'; but as set forth above, the *repairing* and *deployment descriptor* limitations have been rendered obvious in view of the capability to automatically update any modified representations of said descriptor, and in view of a need to implement a proper Descriptor for a deployment as taught by WLS_6.1.

5. Claim 2 is provisionally rejected under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claims 5, 10, 17, 24 of copending Application No. 10,772,613(hereinafter '613). Although the conflicting claims are not identical, they are not patentably distinct from each other because of the following observations.

Following are but a few examples as to how the certain claims from the instant invention and from the above copending application are conflicting with each other.

As per instant claim 2, '613 claim 5 recites generating a representation of one deployment descriptor, creating of one deployment descriptor, validator for validating said deployment descriptor, generating an syntactic/semantic error, a interactive tool to automatically repairing if one first deployment descriptor is defective; but '613 claim 4 does not recite automatically deploying an application associated with said deployment descriptor. Based on '613 user-editable representation of the component being hierarchically presented for enabling the descriptor to be modifiable, and the teaching of a editor by WLS_6.1, this (automatically) *deploying of an application* as intended by '613 tool for repairing a Deployment Descriptor would have been obvious.

As per instant claim 2, '613 claims 10, 17, 24 also recite generating a representation of one deployment descriptor, creating of one deployment descriptor, validator for validating said deployment descriptor, generating error when detecting an error by a validator when a syntactic/semantic fault occurs. Although, claims 10, 17, 24 do not recite *repairing* and automatically deploying; but based on a validator detecting a error, the repairing of a descriptor in a context of a deployment environment would have a obvious step in order for '613 tool provide efficient editing and runtime update support to effectuate this deployment; and the 'automatically deploying' also obvious in view of intended by '613 tool for dynamically repairing a Deployment Descriptor.

This is a provisional obviousness-type double patenting rejection because the conflicting claims have not in fact been patented

Claim Rejections - 35 USC § 101

6. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

7. Claims 1-14 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

The Federal Circuit has recently applied the practical application test in determining whether the claimed subject matter is statutory under 35 U.S.C. § 101. The practical application test requires that a " useful, concrete, and tangible result" be accomplished. An "abstract idea" when practically applied is eligible for a patent. As a consequence, an invention, which is eligible for patenting under 35 U.S.C. § 101, is in the "useful arts" when it is a machine, manufacture, process or composition of matter, which produces a concrete, tangible, and useful result. The test for practical application is thus to determine whether the claimed invention produces a "useful, concrete and tangible result".

The current focus of the Patent Office in regard to statutory inventions under 35 U.S.C. § 101 for method claims and claims that recite a judicial exception (software) is that the claimed invention recite a practical application. Practical application can be provided by a physical transformation or a useful, concrete and tangible result. The following link on the World Wide Web is for the United States Patent And Trademark Office (USPTO) policy on 35 U.S.C. §101.

<http://www.uspto.gov/web/offices/pac/dapp/opla/preognotice/guidelines101_20051026.pdf>

Specifically, claim 1 recites a system comprising a parser, a generator for deploying an application; and there is not sufficient teaching for this system to convey existence of hardware support to effectuate the functionality of the *parser* or the *generator*, all of which being described as mere software entities included in the tool of Figure 2 of the Disclosure. Lacking hardware support in order to carry out the functionality as mentioned above, the claim amounts to mere descriptive listing of functional entities, which amount to not being able to realizing this functionality and yielding thereby a real-world tangible, and useful result. The claim is rejected as non-statutory for not fulfilling a Practical Application result as set forth in the above Guidelines *pdf* file (see 'Functional Descriptive Material', section: Annex IV(a), pg. 53-54).

Claims 2-7 are also rejected for not remedying to the hardware support deficiency of the base claims.

Claim 8 recites a system comprising a parser, a generator and a builder, and for just reciting Functional Descriptive material, the claim is likewise rejected as non-statutory as set forth for claim 1. Claims 9-14 are also rejected for not providing hardware support to embody the software entities of the base claims.

Claim Rejections - 35 USC § 112

8. The following is a quotation of the first paragraph of 35 U.S.C. 112:

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

9. Claims 3, 14, 20-21, 27-28 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the written description requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention.

Claim 3 recites 'the IDE can navigate an user interface to the source of the error'; there no support of this navigating step to the source of an error by a explicit 'IDE' any where in the Specifications.

Claims 14, 20, 27 recite 'modify information in the second representation via the IDE'; there is no corresponding description in the Disclosure about user's usage of a 'IDE' to perform such information modifying.

Claims 21, 28 recite 'IDE capable of generating the second representation'; there is no mention of an IDE in the Specifications in conjunction with the *second representation.*

One of ordinary skill in the art would deem that the inventor does not possess a IDE with the above capabilities as recited at the time the invention was made, as a consequence of explicitly clear, and deliberate description supporting such IDE limitation.

All of the above references to a 'IDE' will be treated as though some part of the (Deployment) builder is performing the recited steps with broadest interpretation.

10. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

11. Claims 3, 14, 20-21, 27-28 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. Claims 3, 14, 20-21, 27, 28 recite the limitation "the IDE". There is insufficient antecedent basis for this limitation in the claim. The use of acronym without a full spelling thereof would also be considered indefinite terminology. This limitation will be treated as a representation within a framework, an editor or a graphical builder.

Claim Rejections - 35 USC § 103

12. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

13. Claims 1-28 are rejected under 35 U.S.C. 103(a) as being unpatentable over WebLogic Server 6.1: *Developing Weblogic Server J2EE Applications*, 10/22/2001, pp. 1-20 <http://web.archive.org/web/20011022014739/edocs.bea.com/wls/docs61/programming/environment.html> (hereinafter WLS_6.1).

As per claim 1, WLS_6.1 discloses a system for automatically maintaining at least one deployment descriptor, comprising:

a parser capable of generating a representation (e.g. *Console* – pg. 9 bottom → top pg. 10; *navigate tree* - pg. 14-16 – Note: console including tree enabling editing of a Descriptor reads on generating representation thereof on console) of the at least one deployment descriptor; a generator capable of creating the at least one deployment descriptor (e.g. XML editor, pg. 9-10; pg. 14-16);

a validator capable of validating the at least one deployment descriptor (e.g. *click Validat*e – pg. 16, item 10, top; *click Validate* – pg. 17 item 10); and wherein the system is capable of automatically deploying an application associated with the at least one deployment descriptor (sec *Web Applications* – pg. 16; sec *Enterprise Applications* – pg. 19).

WLS_6.1 does not explicitly disclose wherein the system is capable of automatically repairing a first deployment descriptor of the at least one deployment descriptors if the first deployment descriptor is defective. WLS_6.1 teaches editor using a form of validator to delete some tree entries in order to persist a confirmed or accepted descriptor into a target application (see substep items 9-11, pg. 17; substep items 8-11, pg. 18-19). In view of the role played by a validator to detect a syntactic or rule-infringing error by the construct of the Descriptor, it would have been obvious for one of skill in the art at the time the invention was made to implement the WLS_6.1 validator in such a way to apply the rule checking as set forth above in order to automatically correct any error (repair a non-compliant Descriptor) by a deployment Descriptor before persisting its corrected version thereof into the Application for deployment, and this is consistent with WLS_6.1 (see pg. 5, substep 7; *Java Compiler, Development Weblogic Server* - pg. 10) utilization of Deployment Descriptor Console to support deploying a enterprise/Web application.

As per claim 2, the limitation as to generating an error (by the validator) when it encounters a syntactic or semantic fault in the at least one deployment descriptor, although not explicitly taught by WLS_6.1, has been rendered obvious in light of the rationale in claim 1.

As per claim 3, WLS_6.1 discloses in response to a user action, the IDE can navigate (e.g. substep 7, pg. 15; substep 8, pg. 17, pg. 18; substep 5, pg. 19) a user interface to the source of the error.

As per claim 4, WLS_6.1 discloses wherein the generator is capable of producing the at least one deployment descriptor from at least one source code file (e.g. XML editor, schema - pg. 14, middle; *web.xml weblogic.xml files* - substep 4, pg. 2).

As per claim 5, WLS_6.1 discloses builder component capable of automatically updating the at least one deployment descriptor to reflect one or more changes (e.g. Using the Administration Console … Editor … click the Persist button - pg. 14) in at least one source code file.

As per claim 6, WLS_6.1 discloses wherein the representation can include information pertaining to at least one of: a Java.TM. archive (JAR), a Web Archive (WAR), an Enterprise Archive (EAR), and a Java.TM. Connector Architecture Component (RAR) – see pg. 2, step 4 to step 7; steps 5-7, pg. 5; *Resource Adapter .rar* - pg. 6, pg. 8).

As per claim 7, WLS_6.1 discloses wherein the at least one deployment descriptor can be expressed as an Extensible Markup Language document (refer to claim 4- see *ejb-jar.xml* - pg. 14-15).

As per claim 8, WLS_6.1 discloses a system for automatically maintaining at least one deployment descriptor, comprising:

a parser capable of generating a first representation of the at least one deployment descriptor; a generator capable of creating a second representation of at least one deployment descriptor based on one or more source files (see substeps 2-5 - pg. 14-15 – Note: XML-parsing with GUI-based tree creation of nodes with each representing a descriptor reads on first and second representation);

all of which having been addressed in claim 1.

WLS_6.1 does not make it explicit that the system is capable of automatically repairing a first deployment descriptor of the at least one deployment descriptors if the first deployment descriptor is defective; but this repairing limitation has been addressed in claim 1.

WLS_6.1 does not explicitly disclose a builder capable of comparing the first representation with the second representation; wherein the builder is capable of updating the first representation based on the second representation if the first representation is out-of-date. However, WLS_6.1 mention about use of a console to create descriptor for deploying EJB or packages to be compiled, which compilation entails version compatibility for the Jar files being deployed using the Console (see pg. 12, middle) as well as verifying version compatibility for non-BEA developers otherwise (see *runtime version* - Web Browser; *software compatibility* - Third Party Software, pg. 11). Since the Jar files encompass retrieval of persisted information for creating deployment descriptor, the necessity as to check on up-to-date version information for those source files is strongly suggested from the above builder environment, e.g. comparing their respective version-controlled identification. Based on well-known concept that developing code based on files fetched for enlistment in a builder (as purported by WLS_6.1) would require that compatibility of files or properties related to deployment are met such as mentioned above, it would have been obvious for one of ordinary skill in the art to implement WLS_6.1 builder with capability to make use of XML Jar files and ensuring that the derived descriptor are also compatible with the deployment of the J2EE Java applications as contemplated above. One would be motivated to implement an update capability in WLS_6.1 builder using its editor (see Editing Deployment Descriptors – pg. 13-15) in that out-of-date (with respect to version compatibility as taught above) console representation of such derived descriptor would be detected via comparison among their respective versioned identity, thus so for the updating step to correct any such out-of-date Descriptor via the Console as set forth in claim 1, using the repairing capability to readjust such a detected error for the reasons as set forth therein.

As per claim 9, refer to claim 4 for descriptor generated from one source code file.

As per claims 10-11, refer to claims 6-7, for respective rejection.

As per claims 12-13, WLS_6.1 discloses wherein information is not deleted from the first representation, and information in the second representation that is not in the first representation is added to the first representation (substep 7, pg. 15; *add elements* - substep 5, pg. 16 – Note: adding one element from a second representation to a first representation reads on the latter representation not deleted).

As per claim 14, refer to claim 3 for the IDE wherein a user can modify information in the second representation.

As per claim 15, WLS_6.1 discloses a method for updating at least one deployment descriptor, comprising:

creating a first representation of the at least one deployment descriptor;

creating a second representation of a second at least one deployment descriptor based on one or more source files;

all of which being addressed in claim 8.

But WLS_6.1 does not explicitly disclose comparing the first representation with the second representation; and updating the first representation based on the second representation if the first representation is out-of-date. However, the above has been addressed as obvious in light of the rationale as set forth in claim 8.

As per claims 16-17, refer to claims 6-7, respectively.

As per claims 18-19, refer to claims 12-13.

As per claims 20-21, refer to claim 14 and claim 13 (Note: traversal of tree with creating, adding, and deleting reads on modifying second representation while generating it in a IDE).

As per claim 22, WLS_6.1 disclose machine readable medium having instructions stored thereon that when executed by a processor cause a system to:

create a first representation of the at least one deployment descriptor; create a second representation of a second at least one deployment descriptor based on one or more source files; compare the first representation with the second representation; update the first representation based on the second representation if the first representation is out-of-date;

all of which having been addressed in claim 15.

As per claims 23-24, refer to claims 6-7, respectively.

As per claims 25-26, refer to claims 12-13.

As per claims 27-28, refer to claims 20-21.

Conclusion

14. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Tuan A Vu whose telephone number is (571) 272-3735. The examiner can normally be reached on 8AM-4:30PM/Mon-Fri.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng-Ai An can be reached on (571)272-3756.

The fax phone number for the organization where this application or proceeding is assigned is (571) 273-3735 (for non-official correspondence - please consult Examiner before using) or 571-273-8300 (for official correspondence) or redirected to customer service at 571-272-3609.

Any inquiry of a general nature or relating to the status of this application should be directed to the TC 2100 Group receptionist: 571-272-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Tuan A Vu
Patent Examiner,
Art Unit 2193
August 27, 2007